



An adaptable and personalised E-learning system applied to computer science Programmes design

Eiman Aeia¹ · Farid Meziane¹ 

Received: 30 May 2018 / Accepted: 5 November 2018 / Published online: 28 November 2018
© The Author(s) 2018

Abstract

With the rapid advances in E-learning systems, personalisation and adaptability have now become important features in the education technology. In this paper, we describe the development of an architecture for A Personalised and Adaptable E-Learning System (APELS) that attempts to contribute to advancements in this field. APELS aims to provide a personalised and adaptable learning environment to users from the freely available resources on the Web. An ontology was employed to model a specific learning subject and to extract the relevant learning resources from the Web based on a learner's model (the learners background, needs and learning styles). The APELS system uses natural language processing techniques to evaluate the content extracted from relevant resources against a set of learning outcomes as defined by standard curricula to enable the appropriate learning of the subject. An application in the computer science field is used to illustrate the working mechanisms of the APELS system and its evaluation based on the ACM/IEEE computing curriculum. An experimental evaluation was conducted with domain experts to evaluate whether APELS can produce the right learning material that suits the learning needs of a learner. The results show that the produced content by APELS is of a good quality and satisfies the learning outcomes for teaching purposes.

Keywords E-learning · Personalised E-learning · Learning styles · ACM/IEEE computing curriculum · Ontology · Natural language processing · Keyword extraction · Key phrases · Dependency relation · Parse tree · Linguistic methods

✉ Farid Meziane
f.meziane@salford.ac.uk

Eiman Aeia
E.aeia@edu.salford.ac.uk

¹ School of Computing, Science and Engineering, University of Salford, Salford M5 4WT, UK

1 Introduction and motivation

1.1 Introduction

Teaching and learning are greatly influenced by the development of Information and Communication Technologies (ICTs) and advanced digital media. Learning using these new media is often referred to as E-Learning (Anii et al. 2017). In this paper, E-Learning is used in the specific context of technology mediated distance learning where technology is used to design and deliver learning materials. Traditional and early E-Learning systems were usually based on static contents. Their design and implementation were unlikely to change and respond to learners needs and preferences (Benhamdi et al. 2017) as the same learning resources are provided to all learners (Halawa et al. 2015). Indeed, learners may have different interests, level of expertise and learning styles. More recent E-Learning systems have attempted to address these issues.

Personalised Learning Environments (PLEs), for example are designed to allow learners to take control of their learning process and experience (Mödrischer 2010). Their aim is to offer to each individual learner, the content that suits better his/ her learning style, background and needs. For example, recommender systems have been used in the development of the NPL-eL E-Learning system (Benhamdi et al. 2017) where through a series of questionnaires and pretesting, the system develops a content, from already predefined contents, that is suitable for individual learners. However, the approach used to identify the learning preferences of individual learners was done in an ad hoc manner and did not take into consideration the advancements in the education and pedagogical fields where different methods are developed and used to identify the learners learning styles and preferences. Furthermore, the content is selected from pre-selected material by the teachers limiting the choice for the many different learning styles available in the wider learning communities.

Another development in the education field and learning in general is the World Wide Web (WWW) which is becoming the premier source of information for many learners. Indeed, there are thousands of lectures, videos, tutorials and books available for use. Unfortunately, with the exponential increase in the number of available resources, most users are spending more time searching, filtering and testing the resources before they can find those satisfying their needs. Hence, there is a need for a new family of E-Learning systems that will address some of these issues that are currently hindering a better use of the current systems and also to take advantage of the available resources and adapt them for the needs of individual learners. The purpose of this paper is to present the development and implementation of the Adaptable and Personalised E-Learning System (APELS) that aims to extend the current understanding and use of conventional E-Learning systems, by using freely available resources on the Web to design and deliver content for individual learners. The APELS system identifies an initial learning style of the learners based on well-known and used methods and then adapts during the learning process. Furthermore, APELS develops the contents based on recognised curricula and assesses the suitability of the designed content based on pre-defined learning outcomes. Hence, the research question is “Can the APELS system produce suitable learning material that suits the learning needs of a particular learner as a teacher would do?” The research question is attempting to evaluate the quality of the content generated by the APELS system by teachers rather

than looking at the learning experience of the students that is outside the scope of this research.

1.2 Research contribution

The main contributions of this research can be summarised as follows:

- A generic architecture is defined for the development of personalised and adaptable E-Learning systems. The architecture can be implemented for various disciplines with the change of only few components, namely the ontology used to model the knowledge of the field and a standard curriculum that will be used to organise the different learning units as required by the discipline.
- To allow a wider coverage of the content of the discipline and satisfy the needs of individual learners, the system uses freely available resources on the WWW.
- The design of the contents for individual learners is based on standard and recognised curricula within the discipline to allow consistency and quality of the learning resources.
- The identification of the learners learning styles is obtained by using known and recognised pedagogical methods.
- The learning outcomes for individual learning units are used to validate and verify that a suitable material is selected during the development of the resources.
- To validate the feasibility of the proposed framework, we use it to develop a sample of computer science modules. The choice of this domain is mainly influenced by the expertise of the authors of this research, the availability of colleagues for the validation of the generated contents and the wider availability of computer science related resources on the WWW. We acknowledge that this could be a more challenging task for disciplines such as sociology and international studies where the resources are scarce and not supported by internationally recognised and adopted curricula.

1.3 Research scope and limitations

The framework developed in this research is the first of its kind to attempt to develop programs' contents from freely available resources on the WWW using knowledge engineering approaches and supported by internationally recognised curricula. Furthermore, the framework is implemented to illustrate how it would work in practice by developing a set of computer science related modules. However, there are some elements of the framework that are not implemented at this stage of the research and these are left for consideration for future developments of the system. Specifically, the following have not been fully addressed:

- Ideally, the work should have been validated by both academics and students to assess the suitability of the content produced by the APELS system. However, an attempt to validate the content by undergraduate students did not lead to satisfactory and usable results. Most students were not sure if the content is suitable for them or not and thought they can learn from it. They did not provide the depth required in

the evaluation of the system. Furthermore, in our views, this need to be validated over a period of time where a group of student will learn from the content produced by APELS and another from the traditional classroom setting. This was not possible during the development of the APELS system. The main purpose of the system evaluation was to assess the quality of the produced material. Choosing the best teaching material that could suit the learning purposes is always a challenging task for teachers (Ellis 1997). Predictive and retrospective evaluation can be conducted by teachers to evaluate available learning material. Predictive evaluation is carried out by expert reviewers prior to delivering the course based on specific criteria, represented by a checklist on how to achieve the course outcome (Ellis 1997). On the other hand, retrospective evaluation is carried out after the material has been used in a teaching context. After that, a decision is made on whether or not the material has worked for learners. Despite the limitations of predictive evaluation represented by the lack of well-defined formula and a subjective nature (Sheldon 1988), this type of evaluation was employed in this research due to the constraints cited above. Hence, the formulation of our research question (see section 1.1).

- The APELS system included an adaptation phase to assess the content produced for the learner and the associated learning style defined in the early phase of the APELS system. The learning style and content can be adapted according to the answers of four questions. The adaptation system was fully implemented but again because the students were not used in the evaluation of the APELS system, this functionality of our system was evaluated using simulation. We believe that this has demonstrated the functionality of the adaptation process and achieved its purpose.
- We have demonstrated the functionality of the APELS system and its implementation in the design of computer science related modules. This is dictated by the background of the authors, the availability of teachers and lecturers in the department for the validation of the system and the rich resources available on the WWW for the computer science field. The authors are also aware of the availability of an international standard curriculum (ACM/IEEE), and familiar with it as that they have used it in the past. We believe that the framework can be used with other disciplines with only few changes mainly with a different ontology and a general curriculum. We also believe that the contents returned could be enriched with the use of available videos. However, experiments with other disciplines and other multimedia resources is out of the scope of this paper and could form the basis for future experiments.

The remaining of the paper is organised as follows: Section 2 reviews some related works and outlines the different methods used in personalised E-learning systems. Section 3 describes the architecture of APELS which is based on three main models that will form the main components of the system, in details. It also will describe a novel learning outcome validation approach and how it uses linguistic features to extract significant key phrases and keywords related to the pre-defined learning outcomes as defined by the Blooms taxonomy using the ACM/IEEE computing curriculum (Sahami et al. 2013). Section 4 describes the implementation of APELS for the computer science field. Section 5 will discuss the system evaluation to test the research hypothesis from the perspectives of experts. It will describe the setting of the experiment, which includes various phases such as testing the system usability,

evaluating the quality of the produced content, and a general discussion. Finally, Section 6 describes the future work and concludes the paper.

2 Related work

Personalised E-learning systems have attracted great interest in the area of technology based education, where their main aim is to offer to each individual learner the content that suits his or her learning style, background and needs. They have been developed to include a variety of techniques which show contrasting forms of teaching.

The approach known as Intelligent Tutoring Systems (ITS) has been pursued by researchers in education, psychology and artificial intelligence. ITS incorporate built-in expert systems to monitor the performance of a learner and to personalise instructions on the basis of adaptation to the learners' learning style, current knowledge level, and appropriate teaching strategies in E-Learning systems (Phobun and Vicheanpanya 2010). For example, AutoTutor is an ITS developed to help students learn about physics and computer literacy (Cai et al. 2015), at the Institute for Intelligent Systems, University of Memphis. AutoTutor helps students learn by holding a conversation in natural language. It also tracks the cognition and emotions of the student and responds in a manner that adapts to the student needs. The InterBook system, originally proposed by (Eklund and Brusilovsky 1999), is an adaptive tutoring system that uses one specific model of a learners knowledge and applies it in order to provide adaptive guidance, navigation, support and help to the user. As a result, this system determines the educational material that is subsequently made into a set of electronic textbooks. Moreover, the ElmArt system provides intelligent tutoring, which enables support for a Lisp course that ranges from concept presentation to debugging programmes, and was advocated as an on-line intelligent textbook that included an integrated problem-solving environment (Weber and Brusilovsky 2001). Although most of the available personalised E-learning systems use a large number of rules to guide the learners in their learning process, these rules are created for a specific domain and cannot be applied if the domain changes (Brusilovsky 2004).

Incorporating a model of learning style has been considered in a variety of personalised E-learning systems in order to improve the effectiveness of the learning process. An example of an ITS that incorporated a single learning style is the Intelligent System for Personalised Instruction in a Remote Environment (INSPIRE) system (Papanikolaou et al. 2003). INSPIRE utilises the Honey and Mumford's learning style model (Honey and Mumford 1992) and adapts the presentation to the learner based on their learning style in order to create diverse lessons that fit individual learners that would meet their objectives. The learner initially completes the Honey and Mumford style questionnaire where different categories, activist, pragmatist, reflector and theorist are recorded. It is an 80-items questionnaire in order to give a comprehensive analysis of learning style and suggestions for action in more depth, which makes it time-consuming. The Felder-Silverman model (Felder and Silverman 1988) is another learning style model that is used by the Oscar Conversational Intelligent Tutoring System (CITS) (Latham et al. 2014). Oscar used a natural language interface to allow learners to construct their own knowledge through discussions. Oscar CITS mimics a human tutor by detecting and adapting to the students learning style whilst directing the conversational process.

Ontologies are increasingly becoming a popular tool for developing personalised E-learning systems see for example (Yarandi et al. 2012). Their ontology model is built to support adaptive learning describing learners' profiles and is used to categorise language learning materials. The proposed system is self-adaptive in which pre and post testing and activities interactions identifies the pace and topics of the next stage. Likewise, Sudhana et al. (2013) proposed an approach that includes a domain ontology for organizing learning material and learner-model ontology to manage the personalised delivery of learning material. Furthermore, Alani et al. (2003), Cassin et al. (2004) and Zouaq and Nkambou (2008) used the notion of ontology to extract information from the Web for educational purposes. Similarly, in this work, we use an ontology to support our information retrieval system to enable the extraction of the relevant information by providing a more organized and classified information about the domain knowledge.

Given their success in E-Commerce, recommender systems have also been used lately in personalised E-Learning systems. The New multi-Personalized Recommender for E-Learning (NPR-el) system (see Benhamdi et al. 2017) integrates a recommendation system in a learning environment to deliver personalised E-Learning. The Cold Start Hybrid Taxonomy Recommender system is used to overcome the cold-start problem that recommender systems suffer from. This problem occurs when the recommender system does not have enough data on the learner's profile to recommend the right learning material (or in the case of E-Commerce system, to recommend the appropriate products and services). The students start by answering a questionnaire that includes questions on their domain of interest, educational content type and preferences. A predefined list of topics is used for the learners to select and recommend from.

The work proposed in this research expands on the strengths of some of the concepts that have been introduced in previous works and reviewed in previous sections. Personalisation and adaptability play a central role in modern E-Learning systems and this work is a consolidation and generalisation of these concepts. We are proposing first a general framework that could be used for various learning areas and second the use of the rich and freely available resources on the WWW. The personalisation aspect of the APELS system is based on the well-established and used VARK (Visual, Auditory, Reading/writing, Kinesthetic) learning styles (Fleming 2001). The adaptability is a continuous process over the learning life cycle of the learner based on feedback and assessments. The proposed framework is based on an ontology that is used to provide a conceptual knowledge of the domain to be considered by the learner.

3 System architecture

This section will describe the overall architecture of the APELS system which purpose is to deliver recommended learning materials to learners with different backgrounds, learning styles and learning needs. A variety of components (referred to as models in the paper) are developed and techniques used to support the adaptability and personalisation features of APELS. The architecture is based on three main models that will form the basis of the system. The three models are: the learner model, the knowledge extraction model, and the content delivery model as shown in Fig. 1.

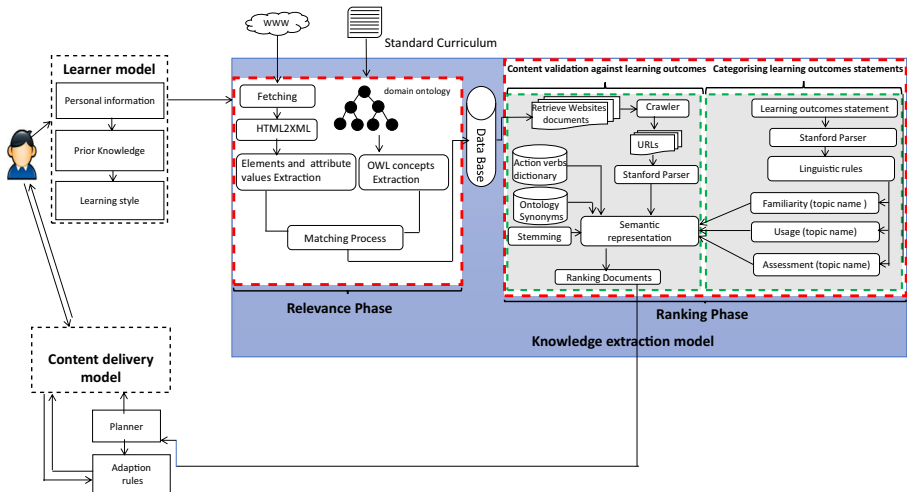


Fig. 1 System architecture showing the different models of the APELS system

3.1 The learner model

First, a learner's model is designed to capture the learner's personal details, learning requirements and the domain they wish to study. The learner model contains all the information about the learner in order to adapt to his or her needs. It contains three components: personal information, prior knowledge and learning style.

3.1.1 Personal information

This component will gather some personal information of the learner such as first name, last name, contact, and address. The learner will then be prompted to enter other information such as their user name and password to create an account in the system.

3.1.2 Prior knowledge

After creating an account, in this step the learner will first choose a specific domain, and then she or he selects a module she or he wishes to study and their level of knowledge (Beginner, Intermediate, Advanced).

3.1.3 Learning style

The VARK (Visual, Auditory, Reading/writing, Kinesthetic) learning styles (Fleming 2001) were chosen in this work to identify an initial learning style of the learners to improve their on-line learning experience. The VARK learning style is found to be relevant as it has an associated tool with the necessary questions that identifies a user's learning style. There are 16 items in the questionnaire as suggested by Fleming regarding the way learners like to learn to analyse their suitable learning styles. After completing the questionnaire, the learners will be informed with their initial learning style preferences as returned by the VARK score. The scores given by the VARK model are a mixture of the

four styles used in the VARK system, namely Visual, Aural, Read/Write and Kinesthetic. The style that obtains the highest score is assigned as the learning style of the user.

3.2 The knowledge extraction model

Once the profile of the learner and his or her chosen area are known, these are saved and submitted to be processed by the knowledge extraction model, which is at the heart of the APELS architecture and is responsible for the extraction of the learning resources from the WWW that would satisfy the user's learning requirements, learning style and preferred learning outcomes. The process in the model is divided into two phases: the relevance phase and the ranking phase as shown in Fig. 1. In the following subsections, we will describe these two phases in details starting first with a high level description, aim and then their implementation details.

3.2.1 Relevance phase

The Relevance phase uses an ontology to help in extracting the required domain knowledge from the WWW in order to retrieve relevant information as per users' requests. A number of functions are developed to support the activities developed in this phase. It starts with fetching the information from the WWW using key words extracted from the learner's domain of interest provided in the learner's model. This is achieved by the fetching function. The contents extracted from the WWW need to be standardised in a common format. In this work, we transform all HTML contents into XHTML and this is performed by the HTML2XHTML function. From the XHTML contents, element and attribute values are extracted and saved in a vector. Similarly, the OWL concepts are extracted from the domain ontology and saved in a different vector. The final function of this phase, the matching process, compares the two vectors containing the XHTML elements and attributes and the OWL concepts to select the best resources for the topic chosen by the learner.

Fetching The relevance phase starts with returning a list of websites that deal with the specific module (learning area). We use Google API (Google 2009), which is implemented in PHP and integrated with the APELS system when fetching specific online content from the WWW based on learners' requests. These websites are first transformed into XHTML to provide the information in an accessible format and easier for extraction and comparison as needed by the other processes. This is followed by the process of transforming HTML content to XHTML that is a better format for the remaining tasks.

Elements and attribute values extraction The specified text enclosed between the start and end tags within XHTML documents are defined as elements such as "Hello, Word!" in <greeting>Hello, world! </greeting>. These elements can be defined using attributes as in (<date="2008-01-10">) where the attribute name is "date" and its value is "2008-01-10". XPath (Clark and Derosé 1999) is utilised in our work to extract XHTML elements and attributes values that are then saved in a vector denoted as $V = [V_1, V_2, V_3, \dots, V_n]$.

Ontology Building our computer science domain ontology is based on the ACM /IEEE computing curriculum, which is internationally recognised and commonly adopted in

the design of computer science related programs across the world (Sahami et al. 2013). The Body of Knowledge (BoK) of ACM/IEEE is organized into a set of Knowledge Areas (KAs) corresponding to typical areas of study in computing such as “Algorithms and Complexity”, “Operating Systems” and “Software Development Fundamentals”. Each KA is broken down into Knowledge Units (KUs). Each KU is divided into a set of topics. Developing existing ontologies that cover computer science domain such as Gašević et al. (2011), Yun et al. (2009), and Rani et al. (2016) would not be efficient since they do not cover all computing areas such as computer engineering, information systems, information technology, and software engineering. The ontology needs to be written in the Ontology Web Language (OWL) that contains concepts or classes that are represented in the computer science curriculum. These concepts will be used to determine similarities with the XHTML elements and attribute values. Therefore, concepts are organised into a set hierarchy, together with the semantic relations that relate them. The Protege editor (Noy et al. 2003) was utilised to develop the ontology. Figure 2 shows a screenshot of the Protege editor illustrating the hierarchy of the relevant domain concepts and their associated relations for the ACM/IEEE computing curriculum.

OWL concepts extraction The OWL file obtained from the Protege tool is uploaded in APELS system to extract the concepts that are represented in a specific domain through the domain ontology. These concepts will be stored in a vector denoted as $C = [c_1, c_2, c_3, \dots, c_n]$ to determine similarities with the XHTML files produced from HTML files.

The matching process The matching process computes the similarity between the ontology concepts that represent the learning domain, saved in the vector C , and the values extracted from the websites, saved in the vector V .

Given a set of relevant online contents and their associated value vectors, the website with the highest similarity is selected as the best matching website for the learner's request. The Dice Coefficient (Dice 1945) was utilised in this process as the similarity measure, as it has been used extensively in many information retrieval (IR) applications due to its good performance and ease of use (Duarte et al. 1999), (Lin 1998). Moreover, the Porter stemming algorithm (Porter 1980) was used in the matching process to improve

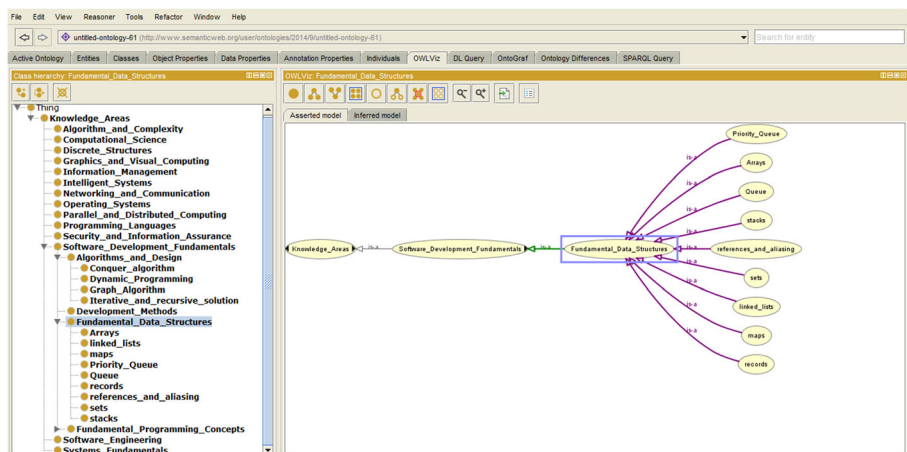


Fig. 2 A screenshot of the computer science ontology

the performance of the similarity measure. In addition, some concepts or terms may be given different names, although they have the same meaning. For instance, the equivalent terms for the concept Calculus includes arithmetic, mathematics etc. This issue was solved by defining corresponding relations such as synonyms in the domain ontology.

Given two vectors C and V defined as:

$C = [c_1, c_2, c_3, \dots, c_m]$ where C_i represent an ontology Concept,
 $V = [V_1, V_2, V_3, \dots, V_m]$ where V_j represent XHTML elements and attribute values extracted.

The similarity measure between vectors C and V using the Dice coefficient is given by Eq. (1):

$$J(C, V) = \frac{2|C \cap V|}{|C| + |V|} \quad (1)$$

Where $C \cap V$ is the number of concepts in C that are also in V (the intersection of the two vectors C and V) and $|C|$ and $|V|$ are the cardinalities of the vectors C and V respectively (the number of elements in C and V respectively). The algorithm developed to measure the similarity between the ontology concepts and XHTML values is given in Fig. 3.

3.2.2 The ranking phase

After the matching process, the learning outcome validation is performed to ensure that the most relevant websites are selected to enable learning according to the learning outcomes set by content specification of the curricula. The validation of learning

Algorithm : Similarity Finding

Input: $C=[C_1, C_2, \dots, C_n]$, $V=[V_1, V_2, \dots, V_m]$

Output: Similarity $S(C, V)$

1. $V = \text{array-unique}(V)$
 2. $S1 = \text{sizeof}(C)$, $S2 = \text{sizeof}(V)$
 3. **Foreach** *Ontology Concepts* C_i **do**
 4. Compare C_i with all values V_j in V
 5. **If** $C_i = V_i$ **then**
 6. $sc = sc + 1$;
 7. **end**
 8. **end**
 9. Compute_Similarity $S(C, V) = \frac{2 * (sc)}{(S1 + S2)}$
 10. Return $S(C, V)$
-

Fig. 3 The similarity measure algorithm

outcomes includes two components: categorising learning outcomes statements and content validation against learning outcomes.

Categorising learning outcomes statements The suitability of the contents of the selected website should be evaluated to ensure that they fit the learner's needs. Matching the content to learning outcomes of curricula is very important when assessing the validity of the selected websites. Basically, learning outcomes are statements of what a student is expected to know, understand and/or be able to demonstrate after the completion of the learning process (Kennedy 2006). Likewise, Mclean and Looker (2006) described learning outcomes as explicit statements of what we want our students to know, understand or be able to do as a result of completing a course. Bloom's Taxonomy (Bloom 1956), is one of the most important and popular frameworks for developing learning outcomes in order to help students understand what is expected of them.

Typically, a learning outcome contains a verb and a noun. In one hand, the verb describes the intended cognitive skill of the Bloom's taxonomy which includes six cognitive levels namely: knowledge, comprehension, application, analysis, synthesis, and evaluation. In the other hand, the noun describes specific subject that student wants to learn. For example: basic structure of the genetic material; nature of chromosomes and the organisation. Furthermore, the Bloom taxonomy identified a list of suitable action verbs to describe each of the six cognitive levels.

In 2001, a former Bloom student, Lorin Anderson, and a group of cognitive psychologists, curriculum theorists and instructional researchers, and testing and assessment specialists published a revision of the Bloom's Taxonomy entitled "A Taxonomy for Teaching, Learning, and Assessment" (Anderson et al. 2001). The revision updates included significant changes in terminology and structure. In the revised framework, "action words" or "verbs", instead of nouns, are used to label the six cognitive levels, and three of the cognitive levels are renamed. Table 4 in Appendix 1 shows the original Bloom's 6 cognitive levels, the revised ones and the action verbs associated with the revised skills.

When defining the learning outcomes for the ACM/IEEE curriculum, although based on the Bloom's taxonomy, a simplified taxonomy was adopted for the computer science field. The terminology was also modified as mastery tasks are used instead of cognitive levels. Three mastery tasks are defined namely: Familiarity, Usage, and Assessment tasks.

- **Familiarity task.** This mastery task concerns the basic awareness of a concept. It provides an answer to the question "What do you know about this?" The initial level of understanding of any topic is answering the question "what the concept is or what it means?" For instance, if we consider the notion of iteration in software development, this would include for-loops, while-loops and iterators. At the "Familiarity task," a student would be expected to understand the definition of the concept of iteration in software development and know why it is a useful technique. This would be the equivalent of the remembering and understanding cognitive levels in the revised Bloom's taxonomy.
- **Usage task.** After introducing a concept to the learner, it would be essential to apply the knowledge in a more practical way, such as using a specific concept in a program, use of a particular proof technique, or performing a particular analysis. It provides an answer to the question "How to use it?" For instance, if we consider

the concept of arrays in programming languages, a student at the usage task, should be able to write or execute a program properly using a form of an array. This would be the equivalent of the applying, analysing and creating cognitive levels.

- **Assessment task.** This task of mastery implies more than using a concept; it involves the ability to select an appropriate approach from different alternatives. It provides an answer to the question “Why would you do that?” Furthermore, the student is able to consider a concept from multiple viewpoints and/or justify the selection of a particular approach to solve a problem. For instance, understanding iteration in software development, at the “Assessment” task would require a student to understand several methods for iteration and be able to appropriately select among them for different applications. This would be the equivalent of the evaluating cognitive level.

The same action verbs used in the Bloom’s taxonomy are used to develop the learning outcomes in the ACM/IEEE curriculum for the three defined tasks.

Content validation against learning outcomes In this research, natural language processing techniques are used to validate the contents against learning outcomes. Linguistic knowledge / features of the words were used to extract significant key phrases and keywords that represent each content, in order to decide which website satisfies the learning outcomes. A number of components are developed to validate the content against learning outcomes. This include a crawler, a dependency relation, and a parse tree.

Crawler: The goal of this step is to return webpages from the website using keywords or topic names. These extracted webpages will be used to validate the content against a set of learning outcomes. An algorithm was developed to check whether the keyword or topic name is included in the URL of the webpage (Meziane and Kasiran 2003). For example, in the website (<http://www.cplusplus.com>) the system will extract all URLs appearing on this website, then the system checks if the keywords or topic name is included in the URL of the webpage, it will save that page in the database to evaluate the content against the identified learning outcomes statements, otherwise it will ignore it, and checks the following webpage and so on. However, some target keywords are not included in the URLs. This issue was solved by extracting the title tag or title element of the webpage, which is a crucial element in identifying the content of the webpage. Then the system checks if the keyword or topic name matches with the text value of the title tag of the webpage. Figure 4 illustrates the crawler process in the APELS system.

Dependency Structures: Dependency Grammar (Tesnière 1959) is a syntactic tradition that determines sentence structure on the basis of word-to-word connections, or dependencies. It names a family of approaches to syntactic analysis that all share a commitment to word-to-word connections. In addition, the document’s words are connected to each other by directed links, and called one of them, the head and the other the dependent. As in the example given in Fig. 5, the dependency link is an arrow pointing from the head (hit) to dependents (Mark, ball) and the arrow pointing from head (ball) to dependents (the).

We employ the Stanford Parser to create a parse tree for a given sentence. For example, the sentence, “algorithm is a list of steps to follow in order to solve a problem” is converted into the parse tree shown in Fig. 6. These structures (parse

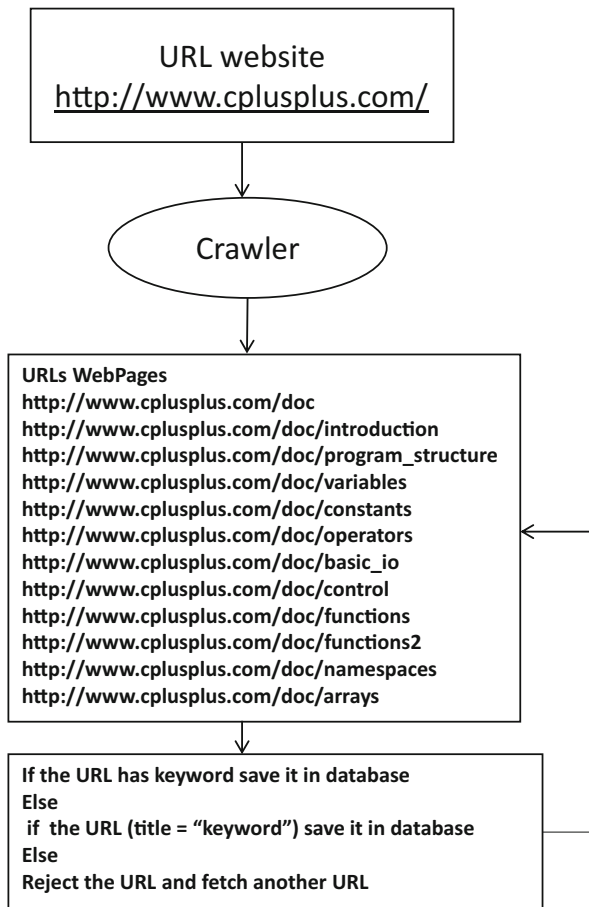


Fig. 4 Crawler process in the APELS system

trees) are then used by the linguistic rules to extract significant key phrases and keywords from the content that would satisfy a specific task.

Linguistic rules: Eight linguistic rules have been designed to capture key phrases and keywords based on determining the linguistic patterns in the dependency relations and parse tree using the Stanford English Parser (Klein and Manning 2003), in order to decide which website satisfies the learning outcomes. They are employed in APELS system to identify the learning outcomes defined in the ACM/IEEE computing

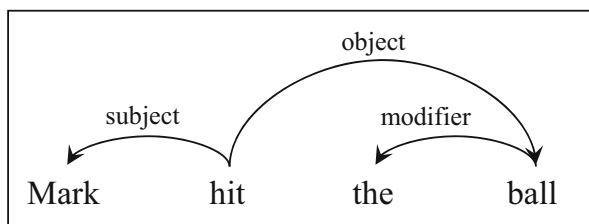


Fig. 5 An example of a dependency structure

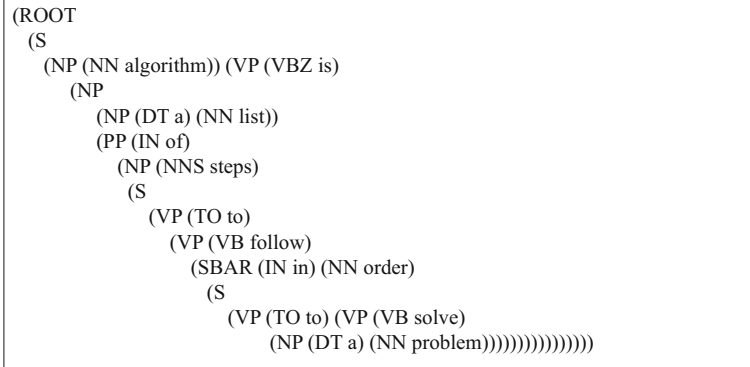


Fig. 6 A sample parse tree using the Stanford Parser

curriculum and these are defined in terms of three tasks: familiarity, usage, and assessment. Each task has an associated set of rules.

- **Familiarity rules.** The first and second rules are employed to extract syntactic structures of sentences that include a noun followed by the verb “to be” expressed as “is” and/or “are”, such as in the phrases “While-loop is” and “Loops are”. These phrases will help a learner to understand what a concept is or what it means.

The first and second rules are used to extract the pattern of the token with the noun tag (NN) in the topic name (algorithm) from the ontology and then check if it is followed by the pattern of token with the verb tag (VB (VBZ is)).

Rule 1. If “ (NN topic name) (VP (vbz is)) “ Then i++;

Rule 2. If “ (NNS topic name) (VP (vbp are))” Then i++;

where i is the number of key phrases appearing in the document.

The third and fourth rules are designed to extract potential relationships between the action verbs associated with the familiarity task and the topic name using dependency relations. Two types of dictionaries are used. The action verbs dictionary that contains the action verbs associated with the familiarity task and the topic name synonym dictionary whose terms are retrieved from the ontology. After parsing each sentence in the document using the Stanford parser, the system extracts the key phrases where the word defined between governor dependency tags, is an action verb associated to familiarity and the word defined between dependent tags is the topic name. Key phrases also can be found in opposite arrangement where the word defined by the governor dependency tags is the topic name and the word defined by the dependent tags is an action verb.

Rule 3. IF “/dep/(governor = actionVerbs[FamiliarityActionverbs] / dependent = topic name[Ontology concepts])” Then j++;

Rule 4. IF “/dep/(governor = topic name[Ontology concepts] / dependent = actionVerbs[FamiliarityActionverbs]) ”; Then j++;

where j is the number of key phrases appearing in the document.

Using expressions such as “For example” or “For instance” in the content will help the reader to understand the content more clearly, instead of providing ambiguous overviews. After parsing each sentence in the document using the Stanford parser, the fifth and sixth rules are used to extract the pattern of token with the noun tag (NN) and then check if the token is “example”.

Rule 5. IF (“NN example”) or (“NN instance”) Then $m++$;

Rule 6. If (“NNS examples”) Then $m++$;

where m is the number of keywords appearing in the document.

- **Usage rules:** Three rules were designed to extract significant key phrases and keywords from the contents that would identify usage tasks. The seventh and eighth rules are utilised to extract the potential relationship between the action verbs associated with the Usage task and the topic name using dependency relations.

Rule 7. IF “/dep(/governor= actionVerbs[UsageActionverbs] / dependent = topic name [Ontology concepts])” Then $n++$;

Rule 8. IF “/dep(/governor= topic name Ontology concepts / dependent = actionVerbs[UsageActionverbs]) ” Then $n++$;

where n is the number of key phrases appearing in the document.

- **Assessment rules:** In this case, the system applies familiarity rules and usage rules for each method or concept. The content produced after applying these rules will help the learner to select the appropriate method or concepts among different methods.

Table 1 shows the rules that are used in the APELS system for extracting key phrases and keywords from contents to decide which website satisfies the familiarity, usage and assessment rules.

Table 1 Rules used to identify familiarity, usage, and assessment tasks

Tasks	Examples of learning outcomes	Rules
Familiarity	Identify and describe uses of iteration?	Rule 1. POS (NN + VB) Rule 2. POS (NNS + VB) Rule 3. Dependency relation (VB, NN) Rule 4. Dependency relation (NN, VB) Rule 5. POS (NN) Rule 6. POS (NNS)
Usage	Write program that use iteration?	Rule 7. Dependency relation (VB, NN) Rule 8. Dependency relation (NN, VB) In addition, Extract instance keyword pattern.
Assessment	Determine which methods of iteration is best or given problem?	Same as above.

3.3 Content delivery model

Once the APELS system has extracted the content taking into consideration the learner's requirements, learning style and learning outcomes, then it will structure and generate a learning plan in a similar way as academic staff would do for their module specification including the contents.

The planner The content delivery model has a planner that structures the produced content into the module title, a summary of the programme, the intended learning outcomes, and the program structure is divided into five categories: topic name, recommended links provided as single links for each topic that provide the personalised content to individual learners according to their prior knowledge and learning style, and learning hours as suggested by the ACM/IEEE curriculum, which was subdivided evenly to cover all the topics. For example two hours for each topic as shown in Fig. 7. The programme structure also includes exercises and evaluations. Figure 7 shows a screenshot of the module specifications page, as produced by APELS, for a specific user in the APELS system for

Module Title: Fundamentals of Programming

Module Code: CF201

First Name: Samir

Last Name: Arohma

Learning Style: ReadWrite

Aims of the Module

The aim of this module is to introduce the students to the fundamentals of programming concepts and enhance their problem solving skills. The students will learn the basics of scalars types (Integer, String, Boolean) and fundamental control structures in procedural programming (Loops, Assignment Statements, Conditional expressions). The module uses the C++ programming language for the implementation. This course will teach the students how to implement file I/O, functions and recursion.

Learning Outcomes

1. Identify and describe used of primitive data types. [Familiarity]
2. Write programs that use primitive data types. [Usage]
3. Write programs that use standard conditional statements. [Usage]
4. Write programs that use iterative control statements. [Usage]
5. Write Programs that use functions. [Usage]
6. Write programs that use file I/O. [Usage]
7. Choose appropriate conditional and iteration constructs for a given programming task. [Assessment]
8. Describe the concept of recursion and give example of its use. [Familiarity]

Program Structure

Topic name	Recommended link	Learning hours	Exercises	Evaluation
Conditional	Link	2	Link	Is this useful?
Loops	Link	2	Link	Is this useful?
Variables	Link	2	Link	Is this useful?
Functions	Link	2	Link	Is this useful?
Recursion	Link	2	Link	Is this useful?

Fig. 7 A screenshot of a module specification as produced by the APELS system

the fundamental programming module including three layered formats consisting of module code, title, aims of the module, intended learning outcomes and program structure.

The adaptation process The planner contains adaptation rules used to modify the learning content based on learners' feedback, and thus, this would be advantageous for the next generation of E-learning systems. A strong feedback from users is a good opportunity to rank and evaluate the content. Accordingly, four questions were devised and implemented in the evaluation section of the module specification page. These are:

- 1- Overall, how satisfied are you with the content?
- 2- Overall, how satisfied are you with the subject coverage of the content?
- 3- Overall, how satisfied are you with the academic quality of the content?
- 4- Overall, how satisfied are you with the learning experience?

Likert five-point Scale is used to records the user's answers where "5" is for strongly satisfied, "4" for Satisfied, "3" for Neutral, "2" for Not satisfied, and "1" for very dissatisfied.

Questions 1, 2 and 3 were designed to investigate the learner's opinion about the quality of the content delivered, whether it is relevant and clear to help the learners to fully comprehend the concepts. Question 4 is associated with the learning style of the learner, which was used to update the learning style based on the learner's feedback.

To evaluate the APELS' produced content by the user, the system calculates the average score of the user's answers to the first three questions using Eq. 2, which helps devise decisions in order to update the content of the links in the module specification.

$$\text{User rating} = \frac{\text{answer}(q1) + \text{answer}(q2) + \text{answer}(q3)}{3} \quad (2)$$

The average score will be stored in the user rating in the learner's model and will be used to update the content of the link in the module specification based on the learner's feedback. The system updates the content of the link by finding the highest score in the user rating which will be recommended to other users with a similar profile. Equation 3 is introduced to calculate the score of the learning style based on the answer to the fourth question. The system first identifies the specific learning style of the learner through the VARK questionnaire (Fleming 2001). This type of learning style can be updated based on the answer to the fourth question of the learner's feedback.

$$\text{LSS} = y - \left(\sum_{i=1}^3 (5 - \text{scoreQuestion}[i]) / 3 \right) \quad (3)$$

Where: LSS is the learning style score, Y is the answer to question 4, i is the answer to questions 1 to 3, and 5 is the number of points on the Likert scale. This score of a particular learning style will be stored and then the planner would automatically update the current one by searching the higher score of a particular learning style.

This aspect of the APELS system is evaluated using a simulation and will be described in the implementation of the system in section 4.

4 The implementation of the APELS system

In the previous section, we have described in details the design of the APELS system. In this section, we show how the APELS system works in practice through the implementation and application of APELS for the development of computer sciences programmes. The choice of this area is mainly dictated by the background of the authors and also the availability of resources both in quality and quantity on the WWW.

To illustrate the implementation of the APELS system, certain elements of the ACM/IEEE computing curriculum were used. For example, the knowledge area “Software Development Fundamentals” module can be defined as a class and its knowledge unites, “Algorithms and Design”, “Fundamental Programming Concepts”, and “Fundamental Data Structures” can be defined as its subclasses. Finally, the lowest level of the hierarchy, which includes a set of topics, can be defined as a subclass of the KU. For example, a set of topics such as “Structure of a Program”, “Variables”, “Expressions”, “Conditional”, “Control Structures”, “Functions”, “File Input and Output”, and “Concept of Recursion” can be defined as subclasses of the KU “Fundamental Programming Concepts”.

APELS returns a list of websites for the Fundamental Programming Concepts module ranking them according to the highest similarity score as shown in Table 2. The Dice similarity measure is used in this research. The closer the score is to 1, the more similar two documents or two contents in general are. In this research, we are not concerned with how high the value is, but which website has the highest similarity.

The results indicate that the websites (www.cal-linux.com/tutorials/) and (www.learn-cpp.org/) have the highest similarities to the OWL file than the other websites. The

Table 2 The matching similarity for fundamental programming concepts module to OWL file

N	WWW	OWL concepts extracted	No of Elements and attribute values extracted	$C \cap V$	$C + V$	$\frac{2 C \cap V }{ C + V }$
1	www.cal-linux.com/tutorials/	8	20	4	28	0.29
2	www.learn-cpp.org/	8	38	5	46	0.22
3	www.penguinprogrammer.co.uk/	8	41	5	49	0.20
4	www.tenouk.com/cncplusplus/tutorials.html	8	52	6	60	0.20
5	www.tutorialcup.com/cplusplus/index.htm	8	55	6	63	0.19
6	www.cplusplus.com/doc/tutorial/	8	57	6	65	0.18
7	www.studytonight.com/cpp/	8	77	6	85	0.14
8	www.w3schools.in/cplusplus/	8	83	6	91	0.13
9	www.cprogramming.pickatutorial.com/	8	107	7	115	0.12
10	www.c4learn.com/cplusplus/cpp-history/	8	99	6	107	0.11
11	www.exforsys.com/tutorials/c-plus-plus.html	8	97	6	105	0.11
12	www.noobtuts.com/cpp	8	29	2	37	0.11
13	www.cprogramming.com/tutorial	8	116	7	124	0.11
14	www.programiz.com/cpp-programming	8	73	4	81	0.10
15	www.functionx.com/cpp/	8	104	5	112	0.09
16	www.tutorialspoint.com/listtutorials/c-and-c++/1	8	156	5	164	0.06
17	www.deitel.com/Tutorials/Freetutorialsand/articles/tabid/1575/Default.aspx#CPLUSPLUS	8	102	2	110	0.04
18	www.en.cppreference.com/w/cpp/language	8	227	4	235	0.03

website (www.cal-linux.com/tutorials/) has a similarity score of (0.29). The ontology concepts of the Fundamental Programming Concepts module saved in vector C is 8, and the values extracted from this website and saved in vector V is 20. The (www.learncpp.org/) also returned a similarity score of (0.22). Therefore, these two websites are the most relevant websites according to the ontology concepts. We note here that the similarity measures are low if compared for example to the thresholds expected in other fields such as information extraction. This was expected as there are usually more terms describing the field of computer science on a website than those describing a subset as extracted from the ontology. The ranking is used to select the highest similarity measure and no thresholds are used. The use of other similarity measures may yield to higher results as only the Dice, Jaccard and Cosine similarity measures were experimented.

5 Experiments and evaluation

In this research, an experimental evaluation was conducted to tests our hypothesis “APELS can produce suitable learning material that suits the learning needs of a learner as teachers would do”. This evaluation was performed by domain experts, who are primarily university academic staff members from various disciplines including computing, mathematics and education, who participated in evaluating the system. To assess the degree to which APELS is successful in achieving its educational objectives requires the testing of the following sub-hypotheses:

The first hypothesis “H1. APELS is usable by the learners and will allow them to provide the right information to determine their backgrounds and needs”. The experiment performed to test this hypothesis involves asking the experts to create an account on the APELS system as if they were learners. This is followed by completing the prior knowledge section and answering the set of the VARK questions to assign the early learning style of the learner. A screenshot of the completion of the first step is given in Fig. 8.

Fig. 8 A screenshot of a user filling the domain selection and a module in the prior knowledge page

The second hypothesis is: “H2. APELS can return suitable learning material based on the background information of the learner.” The main purpose of the system evaluation was to assess the quality of the produced material. Therefore, qualitative methods were applied to gather and analyse the data required for the evaluation system. A questionnaire was designed to elicit information necessary to evaluate the degree to which the experts were satisfied with the content produced by the system, whether it is of good quality and whether or not it satisfies the targeted learning outcomes, namely familiarity, usage or assessment as defined by the ACM/IEEE computing curriculum. The questionnaire incorporates an open comment section whereby the experts can state their opinions concerning the content produced by the system. Some domain experts while testing the system’s usability commented on the system’s interface. Their comments were taken into account to improve the overall usability of the system making it easier and simpler to use for future versions. For instance, in the module specification page, one expert suggested adding certain instructions or explanations to clarify the purpose of each link. Overall, the experts were satisfied with the system interface apart from the weaknesses which were addressed to provide a better interface and experience for future users.

After the experts worked through the system interface, they were asked to assess the quality of the produced content and to indicate whether it satisfied the learning outcome as defined by the ACM/IEEE computing curriculum. While evaluating the quality of the produced content phase, a variety of positive and negative comments were made by the experts. They were specifically about the produced content as related to Familiarity, Usage, or Assessments learning outcomes. Overall, the feedback with regard to matching the content to the learning outcomes was positive. 80% of the experts agree that the provided material was of good quality and that it could be used for preparing and delivering a lecture in order to familiarise the students with a given topic, and even more promising, 90% of them think that the content provided by the system in the experiment was so high in quality that it could be used as teaching material to achieve the Usage task, and 90% of the experts agree that the content satisfied the Assessment learning outcomes as it combines the three types of concepts and provides a simple introduction and fewer examples on each type in order to enable students to compare. They agree that the content was informative and comprehensive and that it clearly reflects the success of the novel learning outcome validation approach and the NLP tool used to perform this function as well the ontology used for information extraction from the WWW. A detailed description of the participants’ backgrounds, experience and expertise is given in Table 5, Appendix 2.

The validation of the adaptation phase was conducted using simulation and a controlled experiment. Four potential virtual learners were used and for each, an account was created and an initial learning style was assigned. Once the contents were developed, we simulated a low average for the answers to the first three questions defined in section 3.3 (The adaptation process) for Learner 1 and Learner 2 and high average for learner 3 and Learner 4. For the answer to question 4, we simulated a high answers for Learner 2 and Learner 4 and low answers for Learner 1 and learner 3. The results of this validation is summarised in Table 3.

Table 3 Validation of the adaptation process

Learner	Initial learning style	Average answers Q1-Q3	Answer Q4	Updated learning resources	Updated learning style
1	Read/Write	2	2	Updated	Updated
2	Read/Write	2	4	Updated	Not updated
3	Visual	4	2	Not updated	Updated
4	Visual	5	4	Not updated	Not updated

When the learner is happy with the content that was created for him or her, or happy with the learning style, these are not changed and will remain the same for the learner and will be assigned for future learner with the same profile. However, if the learner is not satisfied with the content, then from the initially selected list of websites, the next on the list will be selected and used as the new learning material for the learner. This is the cases of Learner 1 and Learner 2 in this experiment. Similarly, as introduced in the planner's section, the learning style is returned as a mixture of the four learning styles but the one with the highest score is assigned as the learning style of the learner. In this experiment, Learner 1 and Learner 3 gave a low answer to question 4. Hence, Learner 1 was moved from the Read/Write learning style to the Visual learning style (the second highest style) and Learner 3 was moved from Visual to Read/Write.

6 Conclusion

An adaptable and personalised E-learning system (APELS) architecture is developed to provide a framework for the development of comprehensive learning environments for learners who cannot follow a conventional programme of study. The system extracts information from freely available resources on the Web taking into consideration the learner's background and requirements to design modules and a planner system to organise the extracted learning material to facilitate the learning process. The process is supported by the development of an ontology to optimise and support the information extraction process. Additionally, natural language processing techniques are utilised to evaluate a topic's content against a set of learning outcomes as defined by standard curricula. An application in the computer science field is used to illustrate the working mechanisms of the proposed framework and its evaluation based on the ACM/IEEE computing curriculum. The APELS system provides a novel addition to the field of adaptive E-learning systems by providing more personalized learning material to each user in a time-efficient way saving his/her time looking for the right course from the hugely available resources on the Web or going through the large number of websites and links returned by traditional search engines. The APELS system will adapt better to the learner's style based on feedback and assessment once the learning process is initiated by the learner. From the evaluation, APELS has received positive comments regarding its overall performance since it has met the main objective of providing

personalised adaptive learning material to E-learners selected from the freely available resources, which successfully meet the pre-defined learning outcomes. From the questioners, the learning material received a positive feedback from the experts that evaluated the APELS system and think that the produced content is of a good quality and that it successfully meets the pre-defined learning outcomes: Familiarity, Usage and Assessment respectively. That clearly reflects the success of the novel learning outcome validation approach and of the ontology tools used for information extraction from the Web.

Similarly, the domain experts praised the adaptability of the system which can change the content based on the user's evaluation. In addition, APELS learns from experience; it updates based on the users feedback. On the other hand, certain issues and problems with the system were highlighted by the domain experts; they were related to the interface of the system, which could be easily updated and rectified. These issues included the font size of some information on the page and the inadequate labelling of the navigation. Furthermore, some experts in the process of the open-feedback phase following the close-questions pinpointed certain weaknesses, such as the search engine that is superimposed by our own ranking system based on keywords and key phrases, so it is vulnerable to misconduct by people who know how the system works.

As future developments, the limitations identified in the introduction section should be addressed with further experiments with different disciplines particularly those that do not have rich resources on the WWW or where the choice is limited. It would be also interesting to include other multimedia contents such as videos and sound tracks to complement the textual information used primarily in this research. Furthermore, it would be interesting to conduct further interviews with some educators in addition to this survey. This would allow to triangulize these findings in a mixed-methods design.

Appendix 1

Table 4 The Bloom original and revised taxonomies and associated action verbs

Original cognitive level	Revised cognitive level	Sample action verbs introducing learning outcomes associated with the cognitive level
Knowledge	Remembering	Define, Identify, Name, Recognize, Retrieve, Duplicate, List, Recall, Reproduce, Tell
Comprehension	Understanding	Calculate, Conclude, Predict, Discuss, Explain, Classify, Clarify, Translate, Reproduce, Exemplify
Application	Applying	Carry out, Demonstrate, Solve, Illustrate, Use, Classify, Execute, Implement, Practice, Utilize
Analysis	Analysing	Discriminate, Compare, Differentiate, Examine, Infer, Attribute, Contrast, Distinguish, Select, Formulate
Synthesis	Evaluating	Check, Judge, Monitor, Critique, Reconstruct, Defend, Verify, Detect, Coordinate, Dispute
Evaluation	Creating	Construct, Design, Compose, Produce, Improve, Create, Invent, Generate, Plan, Combine

Appendix 2

Table 5 Academics evaluators background and experience

Reviewer #	Background and experience
1	A senior Lecturer in computer science and software engineering with over 15 years teaching experience. He teaches various CS modules including networks, operating systems and software engineering. He is a follow of the higher education academy and won various teaching awards. He is also in charge of overlooking programs and modules development in the department. He has used and is an advocate of the use of technology in teaching.
2	A senior Lecturer in computer science and software engineering with over 20 years teaching experience. He teaches programming and data structures & algorithms to undergraduate level. He has also taught mathematics and software quality modules.
3	A lecturer in computer science with over 10 years teaching experience. He teaches a wide range of core modules such as data structures and algorithms, formal development of software systems, and programming. He also has a good experience in the area of machine learning and pattern recognition.
4	A senior lecturer in computer networks with over 15 years teaching experience. He teaches a wide range of core modules including agile software development, software projects with agile techniques, network programming and simulation, software project management, and advanced programming.
5	A senior lecturer in computer science. He is an experienced researcher and developer of virtual environment systems, with a background in both commercial development and academic research and development. He teaches C++ programming, virtual reality and 3D computer graphics.
6	A senior lecturer in data mining and bio-informatics with over twenty years teaching experience. He teaches advanced databases, business intelligence, web semantic and information retrieval.
7	A research assistant in the computer science department. He has a good experience in designing and developing web-based software applications. He also teaches systems analysis, information security, Java programming and C++.
8	A software programmer in the computer science department of a commercial company. He has a good experience and knowledge in software development life cycle, website design and development.
9	A teaching assistant in Computer Science. She has research experience in using ontologies for data extraction. She also has developed database systems for industrial projects.
10	A professor of computer science with over 25 years teaching experience and research. He has taught a large number of computer science modules over his career and is a well-respected researcher in the field of data and knowledge Engineering.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

- Alani, H., Kim, S., Millard, D. E., Weal, M. J., Hall, W., Lewis, P. H., & Shadbolt, N. R. (2003). Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems*, 18, 14–21.
- Anderson, L.W., Krathwohl, D.R., Bloom, B.S. (2001). A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives. Allyn & Bacon. Boston. (Pearson Education Group).
- Anii, K. P., Divjak, B., & Arbanas, K. (2017). Preparing ICT graduates for real-world challenges: results of a meta-analysis. *IEEE Transactions on Education*, 60, 191–197.

- Benhamdi, S., Babouri, A., & Chiky, R. (2017). Personalized recommender system for e-learning environment. *Education and Information Technologies*, 22, 1455–1477.
- Bloom, B. S. (1956). Taxonomy of educational objectives. In *Cognitive domain* (Vol. 1, pp. 20–24). New York: McKay.
- Brusilovsky, P. (2004). KnowledgeTree: A distributed architecture for adaptive e-learning. *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters. May 19–21 2004*. New York, NY, USA, 104–113.
- Cai, Z., Graesser, A., & Hu, X. (2015). ASAT: AutoTutor script authoring tool. *Design Recommendations For Intelligent Tutoring Systems: Authoring Tools*, 3, 199–210.
- Cassin, P., Eliot, C., Lesser, V., Rawlins, K., & Woolf, B. (2004). *Ontology extraction for educational knowledge bases. Agent-Mediated Knowledge Management* (pp. 297–309). Springer Berlin Heidelberg.
- Clark, J., & Derosé, S. (1999). XML path language (XPath) version 1.0. In *w3c recommendation*. <http://www.w3.org/TR/xpath.html>. Accessed 11 Oct 2018.
- Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26, 297–302.
- Duarte, J. M., Santos, J. B. D., & Melo, L. C. (1999). Comparison of similarity coefficients based on RAPD markers in the common bean. *Genetics and Molecular Biology*, 22, 427–432.
- Eklund, J., & Brusilovsky, P. (1999). Interbook: an adaptive tutoring system. *UniServe Science News. March 1999*, 12, 8–13.
- Ellis, R. (1997). The empirical evaluation of language teaching materials. *ELT Journal*, 51, 36–42.
- Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engineering Education*, 78, 674–681.
- Fleming, N. D. (2001). *Teaching and learning styles: VARK strategies*, IGI Global. Christchurch: Author.
- Gašević, D., Zouaq, A., Tomiai, C., Jovanović, J., & Hatala, M. (2011). An approach to folksonomy-based ontology maintenance for learning environments. *IEEE Transactions on Learning Technologies*, 4, 301–314.
- Google (2018). Code search engine. <https://developers.google.com/custom-search/v1/overview>. Accessed 11 Oct 2018.
- Halawa, M.S., Hamed, E.M.R., Shehab, M.E. (2015). Personalized E-learning recommendation model based on psychological type and learning style models. *Intelligent Computing and Information Systems (ICICIS). IEEE Seventh International Conference on*, 2015. IEEE, 578–584.
- Honey, P. & Mumford, A. (1992). The manual of learning styles, available at: <https://www.le.ac.uk/users/rjm1/tutor/resources/learningtheories/honeymumford.html>. Accessed 11 Oct 2018.
- Kennedy, D. (2006). *Writing and using learning outcomes: A practical guide*, University College Cork.
- Klein, D. & Manning, C. D. (2003). Accurate unlexicalized parsing. *Proceedings of the 41st annual meeting of the association for computational linguistics*.
- Latham, A., Crockett, K., & Mclean, D. (2014). An adaptation algorithm for an intelligent natural language tutoring system. *Computers & Education*, 71, 97–110.
- Lin, D. (1998). Automatic retrieval and clustering of similar words. *Proceedings of the 17th International Conference on Computational Linguistics-Volume 2*, 768–774.
- Manning, C. D., Raghavan, P., Schütze, H. (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press.
- McLean, J. & Looker, P. (2006). University of New South Wales Learning and Teaching Unit. Available at: https://teaching.unsw.edu.au/sites/default/files/upload-files/outcomes_levels.pdf. Accessed 11 Oct 2018.
- Meziane, F. & Kasiran, M. K. (2003) Extracting Unstructured Information From the WWW to Support Merchant Existence in E-Commerce, In Antje Dusterhoft and Bernhard Thalheim (Eds), *Lecture Notes in Informatics, Natural Language Processing and Information Systems*, pp.175-185, GI-Edition, Bonn 2003, Germany
- Mödrtscher, F. (2010). Towards a recommender strategy for personal learning environments. *Procedia Computer Science*, 1, 2775–2782.
- Noy, N. F., Crubézy, M., Ferguson, R. W., Knublauch, H., Tu, S. W., Vendetti, J., & Musen, M. A. (2003). Protege-2000: an open-source ontology-development and knowledge-acquisition environment. *American Medical Informatics Association Annual Symposium Proceedings*, 953, 953.
- Papanikolaou, K. A., Grigoriadou, M., Kornilakis, H., & Magoulas, G. D. (2003). Personalizing the interaction in a web-based educational hypermedia system: the case of INSPIRE. *User Modeling and User-Adapted Interaction*, 13, 213–267.
- Phobun, P., & Vicheanpanya, J. (2010). Adaptive intelligent tutoring systems for e-learning systems. *Procedia-Social and Behavioral Sciences*, 2, 4064–4069.
- Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14, 130–137.

- Rani, M., Srivastava, K. V., & Vyas, O. P. (2016). An ontological learning management system. *Computer Applications in Engineering Education*, 24, 706–722.
- Sahami, M., Roach, S., Cuadros-Vargas, E., Leblanc, R. (2013). ACM/IEEE-CS computer science curriculum 2013: Reviewing the ironman report. *Proceeding of the 44th ACM technical symposium on Computer science education*. ACM.
- Sheldon, L. (1988). Evaluating ELT textbooks and materials. *ELT Journal*, 42, 237–246.
- Stanfordparser. (2003). The stanford parser: A statistical parser. <http://nlp.stanford.edu/downloads/lex-parser.shtml>.
- Sudhana, K. M., Raj, V. C., Suresh, R. (2013). An ontology-based framework for context-aware adaptive e-learning system. *Computer Communication and Informatics (ICCCI), 2013 International Conference on*. IEEE.
- Tesnière, L. (1959). *Éléments de syntaxe structurale*, Librairie C. Klincksieck.
- Weber, G., & Brusilovsky, P. (2001). ELM-ART: an adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education (IJAIED)*, 12, 351–384.
- Yarandi, M., Jahankhani, H., Tawil, A.-R. (2012). An ontology-based adaptive mobile learning system based on learners' abilities. *Global Engineering Education Conference (EDUCON), 2012 IEEE*, 1–3.
- Yun, H.-Y., Xu, J.-L., Wei, M.-J. & Xiong, J. (2009). Development of domain ontology for e-learning course. IT in Medicine & Education. ITIME'09. IEEE International Symposium on, 2009. IEEE, 501–506.
- Zouaq, A., & Nkambou, R. (2008). Building domain ontologies from text for educational purposes. *IEEE Transactions on Learning Technologies*, 1, 49–62.